June 22, 2018

RESOURCE ALLOCATION IN 802.11AX NETWORKS

Mukesh Taneja

Bibek Sahu

Ramachandra Murthy

Balamurugan Ramachandran

Ankush Sharma

See next page for additional authors

Inventor(s)
Mukesh Taneja, Bibek Sahu, Ramachandra Murthy, Balamurugan Ramachandran, Ankush Sharma, and Prantik Howlader

RESOURCE ALLOCATION IN 802.11AX NETWORKS

AUTHORS:
Mukesh Taneja
Bibek Sahu
Ramachandra Murthy
Balamurugan Ramachandran
Ankush Sharma
Prantik Howlader

ABSTRACT

Methods of selection of Voice over Internet Protocol (VoIP), video and other users to meet quality of service (QoS) goals and optimize overall performance in 802.11ax networks are provided. These methods allow policy based decisions such as controlling the number of video, VoIP or other users or sub-channel sizes for video (or other) users or deciding data rate (or associated modulation and coding scheme) for each user in each scheduling interval (SI), and allow dynamic decisions for the value of the SI.

DETAILED DESCRIPTION

IEEE 802.11ax supports Downlink/Uplink Orthogonal Frequency Division Multiple Access (DL / UL OFDMA) in addition to other features for High Efficiency wireless local area network (WLAN) operation in dense scenarios. OFDMA supports sub-channels or Resource Units (RUs), where each RU can consist of 26 / 52 / 106 / 242 / 484 / 996 or 2x996 sub-carriers. For example, if a channel bandwidth is 20 MHz, clients could be assigned RUs of sizes 26 / 52 / 106 / 242 sub-carriers (resulting in *approximate* bandwidth allocation of 2 / 4 / 8 / 20 MHz to each client). A client station can be assigned different modulation coding scheme (MCS) values, resulting in different data rates, for each Scheduling Interval (SI) when an access point (AP) uses the IEEE 802.11ax mode of operation to serve clients.

For each (802.11ax) scheduling interval, the AP considers various tasks, such as making a decision on duration of SI itself, selection of suitable client stations to serve for UL / DL, performing client station - RU mapping, assigning suitable values of MCS (and data rates), selecting suitable transmit power values (for AP in DL and suggesting increase / decrease of transmit power values for client stations in UL) and so on. The AP does this while working to achieve various goals such as optimize system capacity, meet Quality of Service (QoS) requirements of different applications and optimize some other fairness objectives. It is a combinatorial optimization problem and good heuristics methods are useful to solve this.

An AP runs resource allocation algorithms that decide various resource parameters for each client. For example, these algorithms decide the stations to serve in each scheduling interval, the RU to be allocated to each selected station, size of each RU, MCS for each RU, transmit power, duration of scheduling interval and so on. These algorithms use various performance indicators (such as channel conditions, network load, QoS provided to each client) which can be captured or derived using current and past observations.

If it can be determined how the network environment is going to evolve in the future, that knowledge can be used to manage resources more effectively. The challenges are how to predict some such parameters, and how to enhance resource allocation methods to manage resources more effectively.

In one embodiment, a method is provided in which parameters such as QoS class indicator (or WLAN Access Class), buffer depth, urgency indicator (a measure of waiting or remaining time of a packet in the AP/WLAN controller system especially for delay sensitive applications), Modulation and Coding Scheme (MCS) index value or signal-to-interference-plus noise (SINR), and location of user (if known), are used to select the following: users to be served in a scheduling interval (SI), RUs to be allocated for selected users, MCS index value for each user, Transmit power of AP for each RU for DL (or factor to control transmit power of each selected station for UL) and duration of Scheduling Interval (SI).

In another embodiment, a resource allocation method is provided in which several parameters in a WLAN network are observed and are used to predict traffic load and a radio resource load indicator, called an RU load indicator. For this wireless resource allocation

method, the RU load indicator captures the impact of stations with demanding requirements such as stations with delay sensitive apps, stations at edge of the cell or stations in bad channel conditions. This information could be obtained periodically, for example every 30 sec or 1 min, or on detection of some events. Linear regression, K Nearest Neighbors (KNN) and Support Vector Machine/Support Vector Regression (SVM/SVR) or other methods are to predict these parameters. The IEEE 802.11ax resource allocation methods are thereby enhanced to use these predicted parameters to help improve efficiency of resource allocation methods.

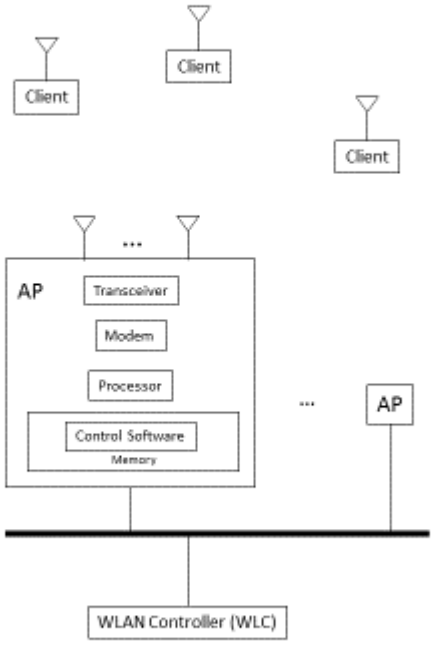

FIG. 1

Referring to FIG. 1, a block diagram is shown of a WLAN that includes a plurality of clients and one or more APs. A WLAN controller (WLC) is in communication with the APs via a wired LAN, for example. An AP includes one or more antennas, a transceiver (that may include multiple transmitters and multiple receivers) that performs radio frequency (RF) transmission and reception functions, a modem that performs baseband modulation and

demodulation functions, a processor (such as a microprocessor or microcontroller), and memory that stores control software.

The memory may comprise read only memory (ROM), random access memory (RAM), magnetic disk storage media devices, optical storage media devices, flash memory devices, electrical, optical, or other physical/tangible memory storage devices. Thus, in general, the memory may comprise one or more tangible (non-transitory) computer readable storage media (e.g., a memory device) encoded with software comprising computer executable instructions and when the software is executed (by the controller) it is operable to perform the operations described herein.

The WLC may be configured with software to perform the operations described herein as being performed by an AP.

The terms "user", "client", "STA" and "station" are used interchangeably in this disclosure.

Appended to this document are Appendix A and Appendix B, which are incorporated as part of this disclosure.

Resource Allocation in IEEE 802.11ax Networks

Reference is made to Appendix A for further details on the resource allocation methods described below.

Method to Select Voice over Internet Protocol (VoIP), Video and other users for a Scheduling Interval (SI):

In this method, the following is performed: Find the maximum number of 26-SC (SubCarrier) RUs possible in a given channel bandwidth. We first consider class I (such as VoIP) apps. We reserve 26 (or 52) SC RUs for VoIP apps (Note: only "number" of RUs. Not necessarily RU indices at this stage). Explaining with 26 SC RU here. If number of VoIP users is more than the number of 26-SC RUs in the given channel bandwidth, we use urgency factor to pick up VoIP users (urgency factor is available for DL apps at AP). Can pick up apps randomly

from that class of apps if urgency factor not available (such as for UL apps) or can pick up based on buffer depth (as buffer information for STAs available at AP for UL communication also).

If some RUs are remaining - we consider class II (i.e. video) applications. We control number of video users and the RU sizes that we allocate to these users in an SI using our method here. We choose number of video users to serve, Nvideo, using a scheme proposed in section 2.2.1 of the attached doc. To pick up this chosen number of video users, i.e. Nvideo (from all the video users): we use urgency factor, buffer depth, MCS values and other parameters as available.

Remaining RUs, if any after above steps, are assigned to class III (and then class IV) applications in our method.

Method for RU Assignment Given a Set of Selected Users:

Once we have selected users to serve in a scheduling interval, we use this method to allocate specific RUs to different users. For a given number of users, there can be several ways to allocate RUs (see Annexure II of Appendix A for an RU table) and we want to choose a suitable RU combination.

In one method, we only consider buffer (depth or) index (of selected users) for selecting RUs. We compute multiplicative sum of RU width and buffer index for all possible combination (of RU allocations) given the user count and select one that gives maximum value. For 40 MHz case, we repeat this computation for NxN combinations where N is the number of RU combinations possible in 20MHz. For 80/160MHz case, we divide number of users to equal subsets and we recursively construct the RU vector.

In another method, we use buffer depth and MCS values to select a specific RU assignment for a given set of (selected) users. Once we have buffer index for a user, we compute the tentative serving time for the user for a given RU, selecting a RU vector combination, and the last MCS reported for that user in that RU. We find the max serving (t_max) time among all the users in the selected combination. Next, we select the RU combination that has minimum t_max for that scheduling interval.

Note that we have already taken MCS into account while choosing users and RUs for them. Once selected, users are served at best possible MCS values (for the corresponding RU at that time). If a set of MCS values are available for a selected RU for a selected user in a given scheduling interval, AP's (or STA's) transmit power can be varied to choose suitable MCS for DL (or UL) transmission.

Methods to Select a Suitable Scheduling Interval:

I: Compute SI dynamically after selecting users / RUs etc. (as in Method I) and keep it bounded by an upper limit as specified by regulatory constraints.

II: Start with default value of 1 ms (or 2 ms) SI and change SI dynamically. Dynamically change SI as follows:

- If it is found that packets for all (or most of) apps can be served within 1 ms and chosen SI = 2 ms (and thus, resulting in very high padding overhead) , change SI to 1 ms.

- If it is found that large number of selected users ready to occupy all resources for SI = 2 ms and they still have many pending bytes in their queues (and we are ok from the point of view of latency constraints), change SI to 3 ms (or even 4 ms). This will allow for dynamic aggregation (for A-MPDU) construction and make system more efficient.

III: We select SI randomly within some (configurable or dynamically computed) thresholds.

In summary, methods are provided for selection of (VoIP, Video and other) users, associated RUs, MCS values, transmit power and other parameters based on buffer depth, urgency indicators, channel conditions, QoS requirements, fairness measures and other parameters to meet QoS goals of delay sensitive users and at the same time, optimize overall performance in 802.11ax networks. These methods allow policy based decisions such as to control number of video users or sub-channel sizes for video (or other) users in each scheduling interval, and dynamic decision for the value of the scheduling interval.

Again, reference is made to Appendix A for more details of the resource allocation methods for 802.11ax networks.

Two example scenarios are shown below. In the first example, as shown in Figures 2-4, the results of the present techniques are compared with a round robin algorithm, each with eight video users. The round robin system does not adequately support users as shown by the higher latencies. Even reducing the round robin system to four users is not sufficient to overcome high latencies.

Figure 2 shows the physical distribution of the various component of the system.

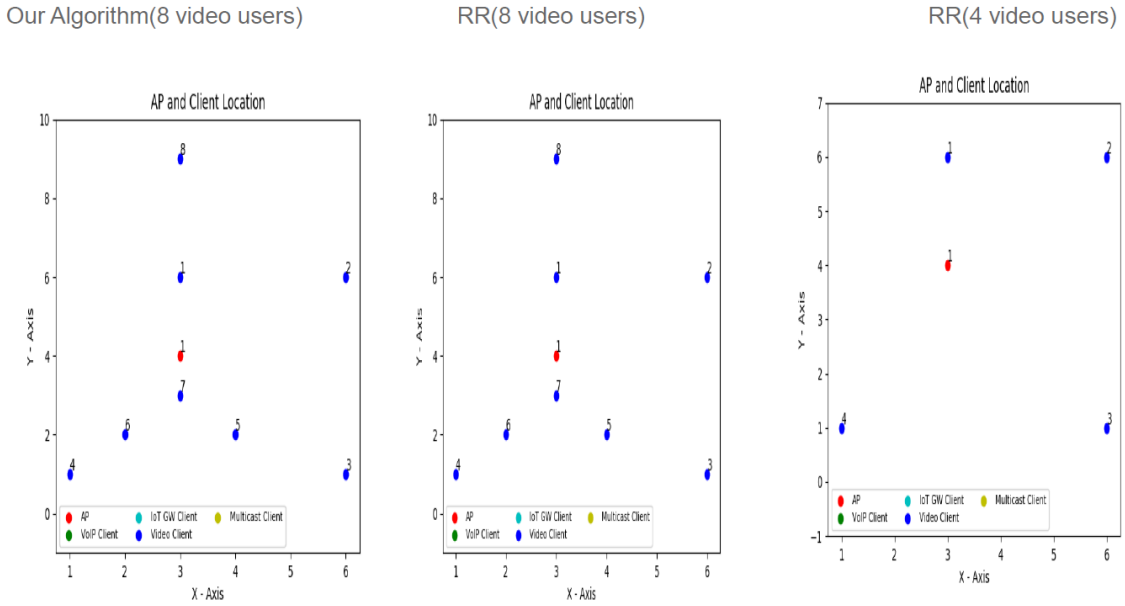

Figure 2

Figures 3 and 4 show throughput and latency of the present techniques as compared to the round robin system. Much higher latencies were observed with round robin as compared to present techniques. Even when the number of users is reduced by 50%, higher latencies are still seen with round robin.

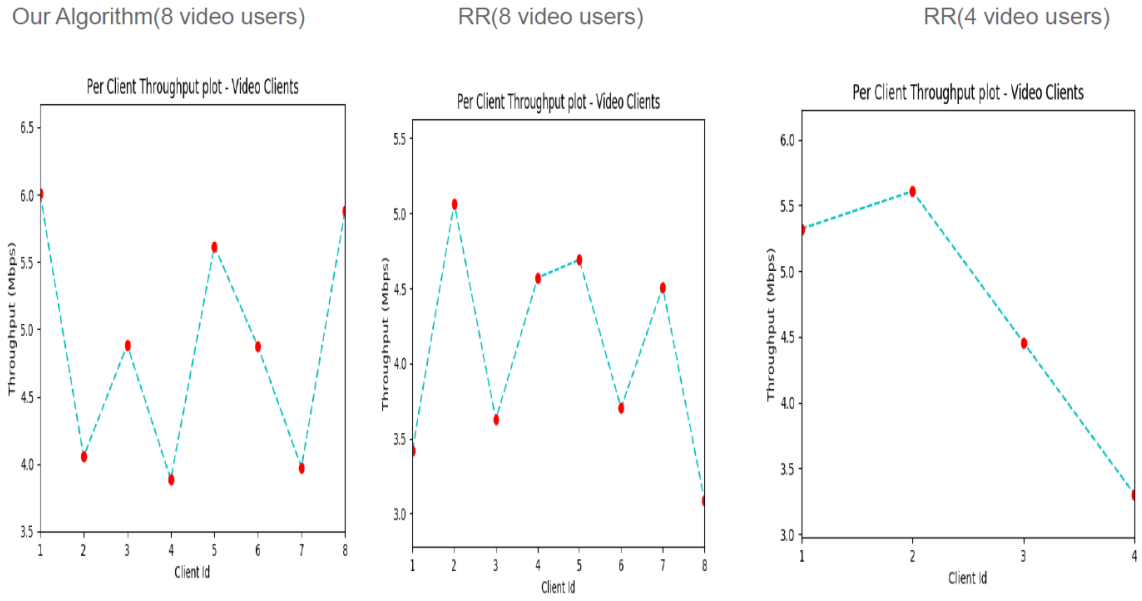

Figure 3

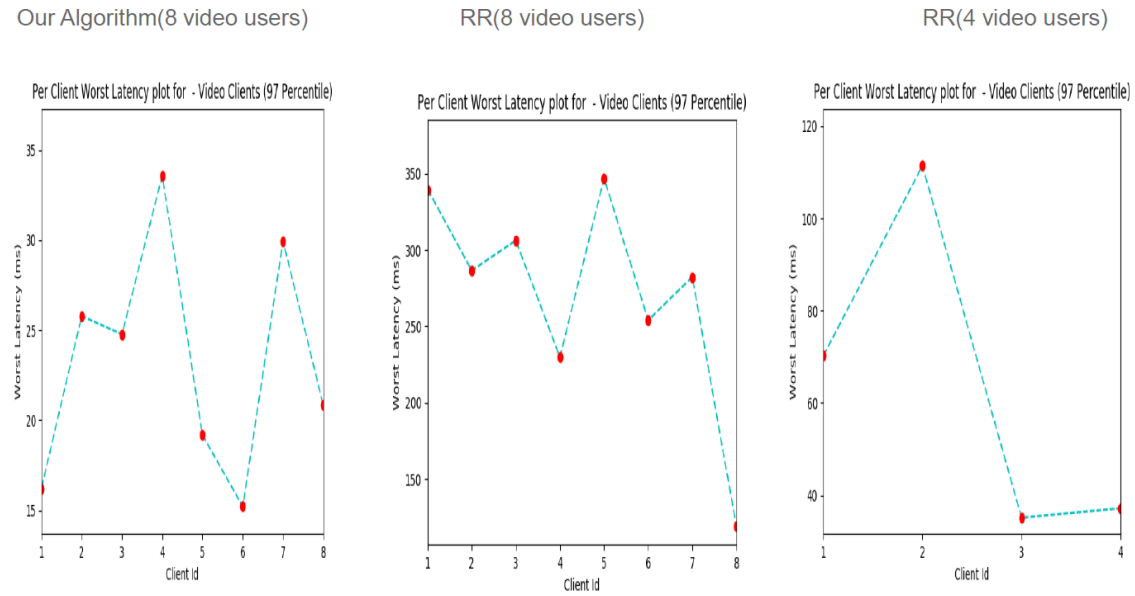

Figure 4

In the second example, as shown in Figures 5-9, the results of the present techniques are compared with an enhanced round robin algorithm. The enhanced round robin system still does not adequately support users as shown by the higher latencies. Even reducing the round robin system to five users is not sufficient to overcome high latencies. The present techniques also

have better delay jitter for VoIP applications and higher throughput for IoT GW users than enhanced round robin techniques.

Figure 5 shows the physical distribution of the various component of the system.

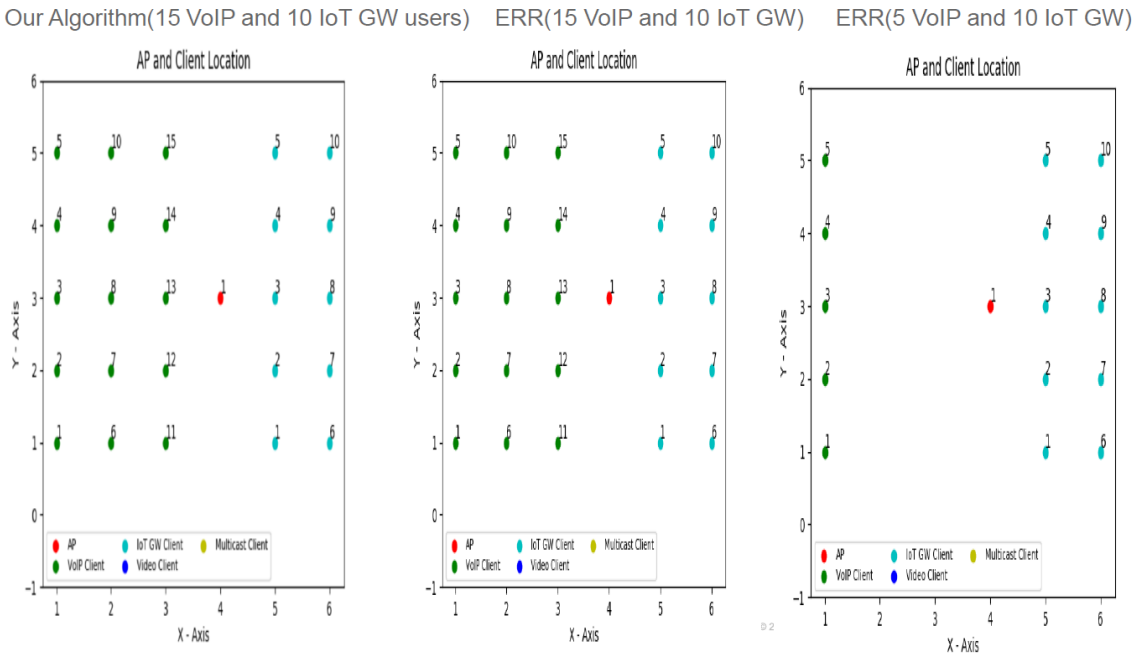

Figure 5

Figures 6-9 show throughput, latency, and jitter results for the present systems versus enhanced round robin systems. Much higher latencies were observed with round robin techniques as compared to present techniques. Even when the number of users is reduced by 50%, higher latencies still remain with round robin.

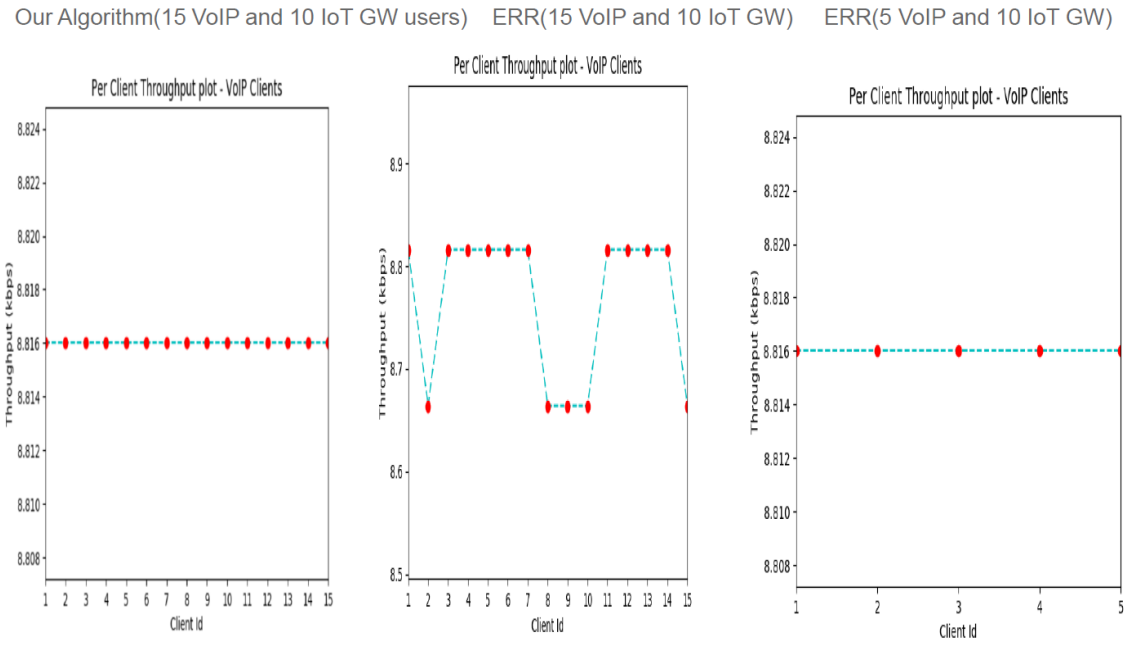

Figure 6

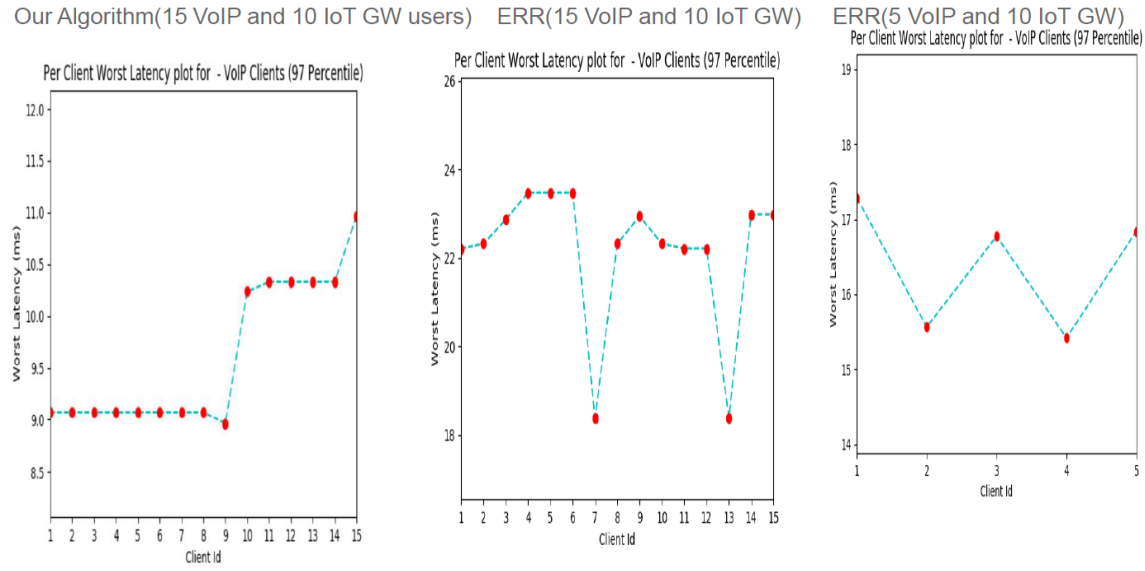

Figure 7

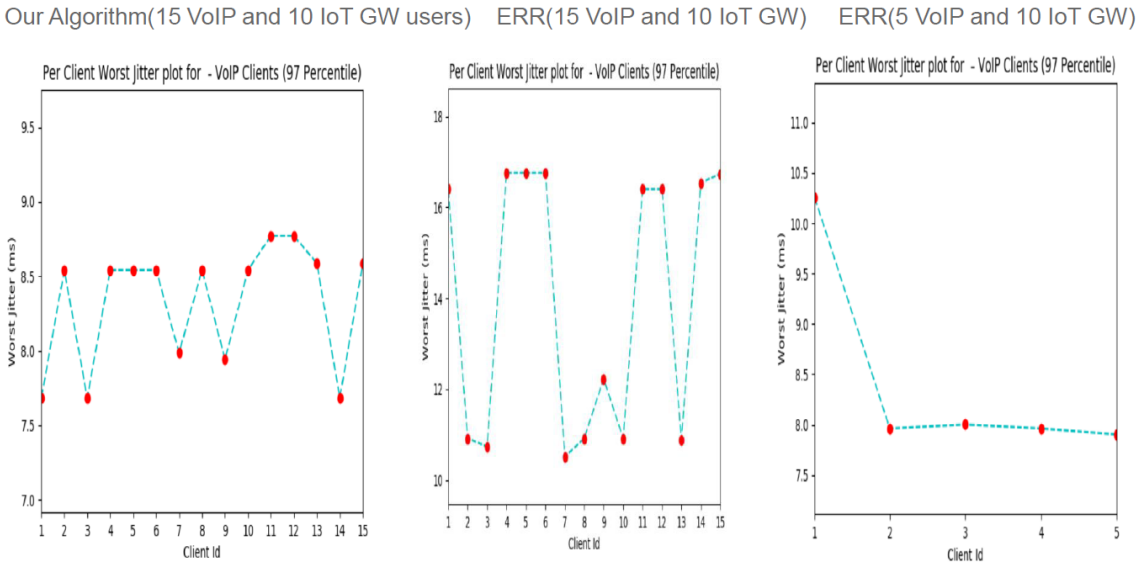

Figure 8

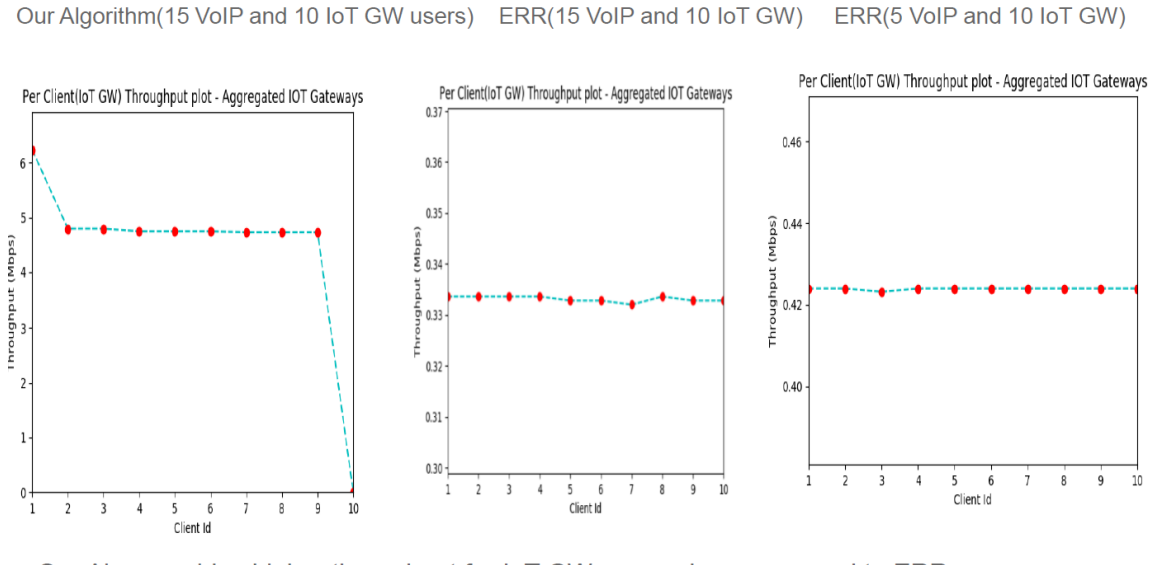

Figure 9

Higher latencies were observed with enhanced RR (ERR) as compared to present techniques. The number of VoIP users were reduced by 66% and still higher latencies with ERR were observed as compared to present techniques. The present techniques provide better jitter bounds for the same number of users. For instance, the number of VoIP users need to be decreased from 15 to 5 (with ERR) to obtain similar jitter bounds. The present techniques

provide higher throughput for IoT GW users when compared to ERR, while providing better delay and jitter bounds for a higher number of VoIP users.

Resource Allocation in 802.11ax Networks Using Predicted Parameters

Reference is made to Appendix A for further details on resource allocation in 802.11as networks using predicted parameters, described below.

An Enhanced Resource Allocation Method using Predicted Parameters:

802.11ax AP's QoS scheduler performs various resource management tasks such as select stations to serve in any scheduling interval, assign RUs to stations, select MCS for each RU and assign transmit power. A Proportional Fair (PF) QoS scheduler at an AP considers instantaneous channel condition for each user and observed throughput for that user to compute a PF metric. It picks up a user with maximum value of PF metric to serve. For an 802.11ax / OFDMA system, it would consider instantaneous channel condition for a given RU and observed throughput for that user to compute a PF metric (for that RU for each user). For each client i, proportional fair metric for each RU c as per a proportional fair scheduling mechanism (at time t when a scheduling decision has to be taken in an 802.11ax system) is given as in Equation 1 of Appendix B.

We define a new metric, H, to be computed at the AP, for user (or client station) i for RU c at time t as given in equation 2 of the attached document. Here, the weight for client station could depend on factors such as buffer length consisting of packets to be transmitted and delay budget within which packets need to be transmitted. In our method, we also include predictive weight for each client station that takes into account predicted parameters at time t in the computation of PF metric H (as shown in Figure 1 and Equation 2 of Appendix B).

We consider following performance indicators for a WLAN AP at any given time instant t:

1. Number of associated users (denoted as N(t)),

2. Number of users who are using delay sensitive applications (such as YouTube video streaming), denoted as Nds(t). A QoS scheduler running at an AP may try to allocate RUs with higher MCS if possible or higher number of RUs or wider RUs to such client stations.

3. Aggregated DL and UL throughput, denoted as AggrT(t), via that AP

4. Location of users: We classify each user to be in one of the three zones: edge, center (e.g. within 2 m of AP) and middle (i.e. between edge and center) of the cell. Let number of associated users at edge, middle and center for this AP be denoted as Ne(t), Nm(t) and Nc(t) respectively. We have, N(t)=Ne(t)+Nm(t)+Nc(t)

5. AP hardware and software loading indicator (HSLI) that takes values as High, Medium or Low.

6. AP RU load indicator (RULI): With this, we capture fraction of total subcarriers that we allocate to users with critical requirements from the point of view of QoS scheduler running on that AP. For example, this could include subcarriers allocated for users with delay sensitive apps or for users who are at edge of the cell or in bad channel conditions or where we are forced to serve at lower MCS to that user. If value of this indicator, RULI, goes up, it may become more challenging to meet QoS requirements of delay sensitive apps or support users in bad channel conditions a dense network.

In a typical deployed scenarios, we may or may not have access to all these performance indicators. We consider only a subset of these indicators to be available to us as input features and we predict other indicators using these. We can capture some of these periodically (say, every minute or every 30 sec) or capture on detection of some events. We derive following indicators using above variables available:

- Net rate at which stations get associated with that AP (denoted as deltaN(t)) and this rate as a fraction of total number of associated users (i.e. deltaN(t) / N(t)),

- Net rate at which stations get added at edge, middle and center zone of a cell (denoted by deltaNe(t), deltaNm(t), deltaNc(t)). We have, deltaN(t)=deltaNe(t)+ deltaNm(t)+ deltaNc(t).

- Change in aggregate traffic load (denoted as deltaAggrT(t)) and this change as a fraction of aggregate throughput, and

- Change in the number of delay sensitive applications during a given time interval (denoted as deltaNds(t))

Once some users have associated with an AP (e.g. when a train arrives near a platform or passengers arrive to board a train), each such user may start one (or more) app(s) in few tens of seconds (or may be already running in some cases). As this number of users changes at different locations in that cell, we use our ML models to predict change in aggregate throughput and RULI that this particular AP is expected to observe in near future. We now define predicted weight for each user using predicted throughput and predicted RULI as in equation 3 of the attached doc.

Note that we presented above for PF scheduler but above predicted parameters and weights can also be used by other QoS scheduler.

Prediction of Traffic Load and RU Load Indicator

We use machine learning (ML) methods such as SVM / SVR, KNN or Linear Regression to predict traffic load and RULI.

We allow use of following combinations of features for predicting aggregate throughout (or change in that). A suitable one can be selected depending on parameters that have been captured in a deployed scenario. If location data of a user has been captured, we recommend use of #8 below.

(Features) : (Target Variable)

1. (Number of users, N) : (Aggregate Throughput, i.e. AggrT at AP)

2. (Number of users at different location, i.e. Ne, Nm, Nc) : (AggrT)

3. (Net rate of change in N, i.e. deltaN) : (AggrT)

4. (deltaN, AggrT) : (Change in aggregate throughput, deltaAggrT)

5. (deltaN, N, AggrT) : (deltaAggrT)

6. (Net rate of change in users at different locations, i.e. deltaNe, deltaNm, deltaNc) : (deltaAggrT)

7. (deltaNe, deltaNm, deltaNc, AggrT) : (deltaAggrT)

8. (deltaNe, deltaNm, deltaNc, N, AggrT) : (deltaAggrT)

Similarly, RULI is predicted using above parameters. If we know the type of applications (such as delay sensitive or non-delay sensitive), that information is also used to predict RULI.

Once we have predicted these parameters, they are used with enhanced resource allocation methods proposed earlier.

Example data with these methods is provided in Appendix B.

In summary, resource allocation mechanisms in 802.11ax/OFDMA type of systems consider some observed parameters and allocate resources as per certain fairness measures. We predict future values of some parameters (such as traffic load and a new parameter, that we call Resource Unit load indicator) using ML methods and enhance resource allocation mechanisms to use these (along with observed values of parameters) to allocate resources more efficiently.

APPENDIX A

Resource Allocation in 802.11ax Networks

Contents

1 Introduction – Problem Definition and Challenges

IEEE802.11ax supports DL / UL OFDMA in addition to other features for High Efficiency WLAN operation in dense scenarios. With OFDMA, it supports sub-channels or Resource Units (RUs) where each RU can consist of 26 / 52 / 104 / 242 / 484 / 996 or 2x996 sub-carriers. As an example, if channel bandwidth is 20 MHz, clients could be assigned RUs of sizes 26 / 52 / 104 / 242 sub-carriers (resulting in *approximate* bandwidth allocation of 2 / 4 / 8 / 20 MHz to each client). A client station can be assigned different MCS values, resulting in different data rates, for each Scheduling Interval (SI) when an AP uses 802.11ax mode of operation to serve clients.

For each (11ax) scheduling interval, AP needs to consider various tasks such as decision on duration of SI itself, selection of suitable client stations to serve for UL / DL, do client station – RU mapping, assign suitable values of MCS (and data rates), choose suitable transmit power values (for AP in DL and suggest increase / decrease of transmit power values for client stations in UL) and so on. It needs to do this while working to achieve various goals such as optimize system capacity, meet QoS requirements of different apps and optimize some other fairness objectives. It is a combinatorial optimization problem and good heuristics methods needed to solve this.

Note: We use “user”, “client”, “STA” and “station” interchangeably in this paper.

2 Methods for Resource Allocation in DL/UL OFDMA based 802.11ax Systems

We propose a method where we use parameters such as QoS class indicator (or WLAN Access Class), buffer depth, urgency indicator (a measure of waiting or remaining time of a packet in the AP/WLC system especially for delay sensitive apps, see Annexure I of this doc), MCS (or SINR) and location of user (if known) to select the following:

- users to be served in a scheduling interval (SI)
- RUs to be allocated for selected users
- MCS for each user
- Transmit power of AP for each RU for DL (or factor to control transmit power of each selected STA for UL)
- duration of SI

2.1 High Level Overview

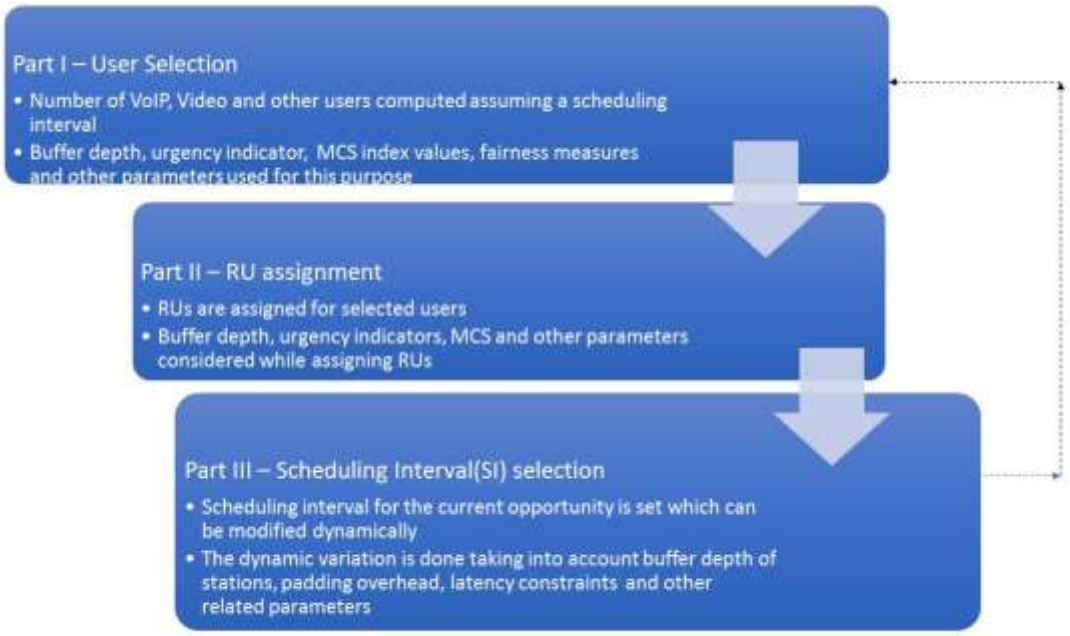

Figure 1: High Level Overview

2.2 Part I – Choosing number of VoIP, Video and other users to serve

In this method, we do as follows:

- We find maximum number of 26-SC (SubCarrier) RUs possible in a given channel bandwidth
    - For example, 9 RUs for 20 MHz or 37 RUs for 80 MHz
- We first consider class I (such as VoIP) apps. We reserve RUs for VoIP apps (Note: only "number" of RUs. Not necessarily RU indices at this stage)
    - 26-SC per VoIP app (can typically serve at least one VoIP packet even at MCS 0 for very small SI)
        - RU index may not be selected at this stage
    - If number of VoIP users more than the number of 26-SC RUs in the given channel bandwidth
        - Use urgency factor to pick up VoIP users (urgency factor is available for DL apps at AP)
        - Can pick up apps randomly from that class of apps if urgency factor not available (such as for UL apps) or can pick up based on buffer depth (as buffer information for STAs available at AP for UL communication also)
    - Can choose 52-SC RU for VoIP if needed

- If some RUs are remaining (after above step for VoIP apps) – we consider class II (i.e. video) apps:
    - We select the number of video users to serve using the method described below:
        - We control number of video users and RU sizes allocated to them in an SI using our method here. For example, we can select smaller number of video users and allocate them bigger RU sizes or vice versa using our method here. Scheme to choose number of video users to serve, Nvideo, is described in the next subsection
        - Number of video users = min(total number of video users with non-zero buffer depth, random number chosen in the next subsection, i.e. Nvideo)
        - To pick up this chosen number of video users (from all the video users)
            - For DL apps: we pick up those video users that have higher value of urgency indicator than other video users. We optionally also allow use of MCS values to select these users
            - For UL apps, we pick up selected number of users using buffer depth (and MCS if available) information
        - Note: Number of such users to be less than or equal to number of 26-SC RUs (after allocating RUs to VoIP users)
    - We make following a configurable parameter:
        - Min number of SCs to be assigned to any video user
- For allotting RU sizes to selected video users, we do as follows:
    - Policy I: Allot RU sizes to selected video users randomly
    - Policy II: Allot RU sizes to selected video users after taking into account buffer depth (and MCS values if available)
    - Policy III: Allot RU sizes using buffer depth, urgency indicator (if urgency factor available such as for DL apps) and MCS values if available
- Remaining RUs, if any after above steps, are assigned to class III apps in our method:
    - Pick up users based on buffer depth
    - Buffer depth can be used to decide RU-size for each class III app
- Remaining RUs, if any after above steps, to be assigned to class IV apps (such as background or best effort apps)

2.2.1 Choosing the number of Video (and other) users to serve

We use this method to select number of video users to serve in a scheduling interval. We use the following notations:

Number of VoIP apps (with non-zero DL queue depth): N_{VoIP}

Number of 26-SC RUs assigned to VoIP apps: RU_{VoIP}^{base}

Number of remaining 26-SC RUs for video (plus other) users: ERU_{video+}^{base}

Date printed: 10/12/2017

We have,

$RU^{base} = 26\,\text{SC RU}$

$\text{RU}_{\text{Total}}^{\text{base}} = \text{Number of } 26\text{-SC RUs in a given channel bandwidth}$

$\text{RU}_{\text{VoIP}}^{\text{base}} = N_{VoIP} * RU^{base}$

$ERU_{Video+}^{base} = RU_{Total}^{base} - RU_{VoIP}^{base}$

Number of video users that we selected to be served: N_{video}

We try to give an RU of size 52-SC (or whatever was configured as min value for video users – 52 / 106 / ….) or higher for any selected video user. (Can give 26-SC RU if only one 26-SC left after allocating to VoIP users).

We use a configurable parameter, factor_video, to influence RU sizes that are allocating for video users (with $0 \le factor\_video \le 1$)

We select number of vieo users to serve, N_{Video}, as a random number in the range [1, max(1, factor_video * ERU_{Video+}^{base})]

For example, factor_video should be configured ≤ 0.5 for the case where we attempt to give at least 52-SC RU to each selected video user. As another example, factor_video = 0 to serve one video user only

Let's say that approximate normalized buffer depth for N_{Video} video users (with non-empty buffer) be: b, n*b, m*b, k*b, ….. Here, n, m, k, …. are integers.

$RU_{video}^{base} \le ERU_{video+}^{base}$

We compute a factor as follows:

$$RU\_fac\_video = floor\left(\frac{ERU_{Video+}^{base}}{1+n+m+..}\right)$$

This factor, RU_fac_video, along with other parameters are used for RU assignment.

Above methods can also applied to class III users.

2.3 Part II – RU Assignment (and MCS selection)

Once we have selected users to serve in a scheduling interval, we use this method to allocate specific RUs to different users. For a given number of users, there can be several ways to allocate RUs (see Annexure II of this doc) and we want to choose a suitable RU combination.

2.3.1 Part II A

In this method, we only consider buffer (depth or) index (of selected users) for selecting RUs. We compute multiplicative sum of RU width and buffer index for all possible combination (of RU allocations) given the user count and select one that gives maximum value.

For each RU combination (for a given user count), we compute a factor, S, as follows (for 20 MHz):

$$S = \sum_{k=0}^{n} RUType^k * BufV^k$$

Here,

n = numUsers = number of users that we selected to serve using methods in previous sub-section) = RU Vector size,
BufV: vector of buffer indices
RUType is chosen proportional to RU width of a RU,
(Note : BufV vector and RU vector are sorted in descending order)

We find the maximum value of S and select that combination. If there exist multiple possibilities, we chose randomly among those

We now extend above method for the case when channel bandwidth is 40, 80 or 160Mhz.

- For 40Mhz case, we repeat above computation for NxN combinations where N is the number of RU combinations possible in 20Mhz.
- For 80/160Mhz case, we divide number of users to equal subsets and we recursively construct the RU vector.

2.3.2 Part II B

In another method, we use buffer depth and MCS values to select a specific RU assignment for a given set of (selected) users. Once we have buffer index for a user, we compute the tentative serving time for the user for a given RU, selecting a RU vector combination, and the last MCS reported for that user in that RU. We find the max serving (t_max) time among all the users in the selected combination. Next, we select the RU combination that has minimum t_max for that scheduling interval.

2.3.3 MCS Selection

We have already taken MCS into account while choosing users and RUs for them. Once selected, users are served at best possible MCS values. If a set of MCS available for a selected RU for a selected user in a given scheduling interval, AP's (or STA's) transmit power can be varied to choose suitable MCS for DL (or UL) transmission.

2.4 Part III – Determination and Dynamic Adjustment of Scheduling Interval

We allow use of different methods to select a suitable scheduling interval.

I: We compute SI dynamically after selecting users / RUs etc. (as in Method I) and keep it bounded by an upper limit as specified by regulatory constraints.

II: We start with default value of 1 ms (or 2 ms) SI and change SI dynamically.

We dynamically change of SI follows:

- If it is found that packets for all (or most of) apps can be served within 1 ms and chosen SI = 2 ms (and thus, resulting in very high padding overhead) , we change SI to 1 ms
- If it is found that large number of selected users ready to occupy all resources for SI = 2 ms and they still have many pending bytes in their queues (and we are ok from the point of view of latency constraints), we can change SI to 3 ms (or even 4 ms). This will allow for dynamic aggregation (for A-MPDU) construction and make system more efficient.

III: We select SI randomly within some (configurable or dynamically computed) thresholds.

3 Abbreviations

MCS (index value): Modulation and Coding Scheme (index value)
RU: Resource Unit
SC: Subcarrier
SI: Scheduling Interval

4 Annexure I – Urgency Indicator

From "Methods for Network Slicing in 802.11ax type of Systems, Mukesh Taneja, https://priorart.ip.com/IPCOM/000250415": A delay budget is assigned for DL packets in the AP / WLAN controller (WLC) subsystem. For example, if end-to-end delay required is 150 ms, a delay budget in the AP / WLC subsystem can be assigned as 75 ms after analyzing end-to-end delay in the deployed network architecture. A remaining time metric is computed for each delay sensitive application by considering its delay budget and amount of time it has spent in the AP / WLC subsystem for first few packets of that application.

For the purpose of notations, it is first assumed that each (client) station i has at most one application that corresponds to a network slice that has stringent delay requirements. This is later extended for the case when a client station may have two or more applications with stringent delay requirements and where each such application may correspond to different network slices. DL traffic is first considered and later extended for UL traffic.

We consider packets pending in the AP queues for each client station i (for network slice k) and identify packets for which either waiting time in the WLC/AP subsystem is above a threshold or remaining time is below a threshold. These thresholds could be pre-specified (or dynamically computed via some policies). Let $len\_thresh_{i,k}(t)$ be the length of packets in the queue for station i corresponding to slice k for which waiting time (in the AP/WLC subsystem) is greater than a threshold, wt_thresh(k), for slice k or remaining time is less than rem_thresh(k) for slice k. Total length of such packets for slice k is given as:

$$len\_thresh_k(t) = \sum_{i:i \in k} len\_thresh_i(t)$$

An urgency indicator is defined for station i belonging to slice k using length of packets that have crossed weighting or remaining time thresholds as above and weighted average of data rate with which the AP had been able to send data to that station as below:

$$urgency\_indicator^{I}_{i,k}(t) = \lambda_{i,k}(t) * \frac{len\_thresh_{i,k}(t)}{dRate\_wavg_{i,k}(t)}$$

Equation 1: Urgency Indicator (of Type I) for client i belonging to slice k

Here, $dRate\_wavg_{i,k}(t)$, is weighted average of data rate with which AP was able to transmit data to client i for data belonging to slice k (as selected by QoS scheduler running at AP). With weighted average, a higher weight can be given to recent values over the values that were observed earlier (i.e. beyond a recent time interval). Also, $\lambda_{i,k}(t)$, is a pre-specified (or dynamically computed) scaling factor for station i belonging to slice k. It can also be used to normalize value of urgency indicators across stations.

An urgency indicator is defined using a different way as follows:

$$urgency\_indicator^{II}_{i,k}(t) = \lambda_{i,k}(t) * \frac{len\_thresh_{i,k}(t)}{MCSmedian\_wavg_{i,k}(t)}$$

Equation 2: Urgency Indicator (of Type II) for client i belonging to slice k

Note that we use two thresholds, urgency_indicator_min and urgency_indicator_max, and ensure the following:

$$urgency\_indicator\_\min^{I}_{k} \le urgency\_indicator^{I}_{i,k}(t) \le urgency\_indicator\_\max^{I}_{k}$$

$$urgency\_indicator\_\min^{II}_{k} \le urgency\_indicator^{II}_{i,k}(t) \le urgency\_indicator\_\max^{II}_{k}$$

Here, $MCSmedian\_wavg_{i,k}(t)$, is weighted average of median of MCS values with which the AP can send DL data to client station i for data belonging to slice k (as reported by client station i to AP after considering MCS values for various RUs).

5 Annexure II – RU Table

Copying / pasting RU table from Table 28-24, IEEE802.11ax D2.0
We need to select a specific RU assignment from large number of combinations (hundreds / thousands) possible from this table.

Table 28-24—RU Allocation subfield

8 bits indices (B7 B6 B5 B4 B3 B2 B1 B0)	#1	#2	#3	#4	#5	#6	#7	#8	#9	Number of entries
00000000	26	26	26	26	26	26	26	26	26	1
00000001	26	26	26	26	26	26	26	52		1
00000010	26	26	26	26	26	52		26	26	1
00000011	26	26	26	26	26	52		52		1
00000100	26	26	52		26	26	26	26	26	1
00000101	26	26	52		26	26	26	52		1
00000110	26	26	52		26	52		26	26	1
00000111	26	26	52		26	52		52		1
00001000	52		26	26	26	26	26	26	26	1
00001001	52		26	26	26	26	26	52		1
00001010	52		26	26	26	52		26	26	1
00001011	52		26	26	26	52		52		1
00001100	52		52		26	26	26	26	26	1
00001101	52		52		26	26	26	52		1
00001110	52		52		26	52		26	26	1
00001111	52		52		26	52		52		1
$00010y_2y_1y_0$	52		52		-	106				8
$00011y_2y_1y_0$	106				-	52		52		8
$00100y_2y_1y_0$	26	26	26	26	26	106				8

Table 28-24—RU Allocation subfield *(continued)*

8 bits indices (B7 B6 B5 B4 B3 B2 B1 B0)	#1	#2	#3	#4	#5	#6	#7	#8	#9	Number of entries
00101$y_2y_1y_0$	26	26	52		26	106				8
00110$y_2y_1y_0$	52		26	26	26	106				8
00111$y_2y_1y_0$	52		52		26	106				8
01000$y_2y_1y_0$	106				26	26	26	26	26	8
01001$y_2y_1y_0$	106				26	26	26	52		8
01010$y_2y_1y_0$	106				26	52		26	26	8
01011$y_2y_1y_0$	106				26	52		52		8
0110$y_1y_0z_1z_0$	106				-	106				16
01110000	52		52		-	52		52		1
01110001	242-tone RU empty									1
01110010	484-tone RU with no User fields in the HE-SIG-B content channel containing this RU Allocation subfield									1
01110011	996-tone RU with no User fields in the HE-SIG-B content channel containing this RU Allocation subfield									1
011101x_1x_0	Reserved									4
01111$y_2y_1y_0$	Reserved									8
10$y_2y_1y_0z_2z_1z_0$	106				26	106				64
11000$y_2y_1y_0$	242									8
11001$y_2y_1y_0$	484									8
11010$y_2y_1y_0$	996									8
11011$y_2y_1y_0$	2×996									8
111$x_4x_3x_2x_1x_0$	Reserved									32

APPENDIX B

Resource Allocation in 802.11ax Networks using Predicted Parameters

Mukesh Taneja, Ankush Sharma, Prantik Howlader, Bibek Sahu, Balamurugan Ramachandran, Ramachandra Murthy

I. ABSTRACT

Wireless systems such as IEEE802.11ax use multi-user OFDMA transmission and support provision of group of subcarriers, called Resource Units (RUs), to different client stations. A resource allocation algorithm in an 802.11ax AP helps to determine resources to allocate to different users in each scheduling interval. Proportional fair and chanel condition aware weighted round robin are some method that have been used for resource allocation in some OFDMA based wireless systems. These methods rely on current and past performance parameters (such as channel conditions, observed QoS, buffer depth and other parameters) to allocate resources to multiple client stations. We propose resource allocation methods where we observe several parameters in a WLAN network and use these to predict traffic load and a radio resource load indicator, that we call RU load indicator. For our wireless resource allocation method, RU load indicator captures impact of stations with demanding requirements such as stations with delay sensitive apps, stations at edge of the cell or stations in bad channel conditions. This information could be obtained periodically, for example every 30 sec or 1 min, or on detection of some events. We use Linear Regression, KNN and SVM/SVR methods to predict these parameters. We enhance 802.11ax resource allocation methods to use these predicted parameters to help improve performance of a WLAN network.

II. AN ENHANCED RESOURCE ALLOCATION METHOD USING PREDICTED PARAMETERS

802.11ax AP's QoS scheduler needs to perform various resource management tasks such as select stations to serve in any scheduling interval, assign RUs to stations, select MCS for each RU and assign transmit power. A Proportional Fair (PF) QoS scheduler (such as the one described by authors in [1] for LTE) at an AP considers instantaneous channel condition for each user and observed throughput for that user to compute a PF metric. It picks up a user with maximum value of PF metric to serve. For an 802.11ax / OFDMA system, it would consider instantaneous channel condition for a given RU and observed throughput for that user to compute a PF metric (for that RU for each user). For each client i, proportional fair metric for each RU c as per a proportional fair scheduling mechanism (at time t when a scheduling decision has to be taken in an 802.11ax system) is given as:

$$M_i^c(t) = \frac{r_i^c(t)}{R_i(t)} \qquad (1)$$

Here, $r_i^c(t)$, is the instantaneous channel condition of user i at time t for RU c and $R_i(t)$ is the long term service rate of user i at time t. It picks up a user with maximum value of PF metric to serve for a given RU. We define a new metric, $H_i^c(t)$, to be computed at AP, for user (or client station) i for RU c at time t as

$$H_i^c(t) = w_i^{predict}(t) * w_i(t) * \frac{r_i^c(t)}{R_i(t)} \qquad (2)$$

Here, weight $w_i(t)$ for client station i could depend on factors such as buffer length consisting of packets to be transmitted and delay budget within which packets need to be transmitted. In the method proposed here, we include $w_i^{predict}(t)$ as the weight for client station i that takes into account predicted parameters at time t in the computation of PF metric H (as shown in Figure 1 and equation 2). We consider following performance indicators for a WLAN AP at any given time instant t:

1. Number of associated users (denoted as N(t)),
2. Number of users who are using delay sensitive applications (such as YouTube video streaming), denoted as Nds(t). A QoS scheduler running at an AP may try to allocate RUs with higher MCS if possible or higher number of RUs or wider RUs to such client stations.
3. Aggregated DL and UL throughput, denoted as AggrT(t), via that AP

4. Location of users: We classify each user to be in one of the three zones: edge, center (e.g. within 2 m of AP) and middle (i.e. between edge and center) of the cell. Let number of associated users at edge, middle and center for this AP be denoted as Ne(t), Nm(t) and Nc(t) respectively. We have, N(t)=Ne(t)+Nm(t)+Nc(t)
5. AP hardware and software loading indicator (HSLI) that takes values as High, Medium or Low.
6. AP RU load indicator (RULI): With this, we capture fraction of total subcarriers that we allocate to users with critical requirements from the point of view of QoS scheduler running on that AP. For example, this could include subcarriers allocated for users with delay sensitive apps or for users who are at edge of the cell or in bad channel conditions or where we are forced to serve at lower MCS to that user. If value of this indicator, RULI, goes up, it may become more challenging to meet QoS requirements of delay sensitive apps or support users in bad channel conditions a dense network.

In a typical deployed scenarios, we may or may not have access to all these performance indicators. We consider only a subset of these indicators to be available to us as input features and we predict other indicators using these. We can capture some of these periodically (say, every minute or every 30 sec) or capture on detection of some events. We derive following indicators using above variables available to us:

- Net rate at which stations get associated with that AP (denoted as ΔN(t)) and this rate as a fraction of total number of associated users (i.e. ΔN(t) / N(t)),
- Net rate at which stations get added at edge, middle and center zone of a cell (denoted by ΔNe(t), ΔNm(t), ΔNc(t)). We have, ΔN(t)=ΔNe(t)+ ΔNm(t)+ ΔNc(t).
- Change in aggregate traffic load (denoted as ΔAggrT(t)) and this change as a fraction of aggregate throughput, and
- Change in the number of delay sensitive applications during a given time interval (denoted as ΔNds(t))

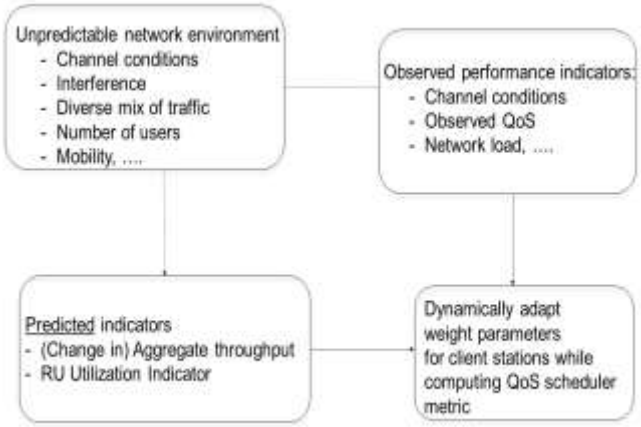

Figure 1: Weight adaptation of QoS scheduler metric using predicted and observed performance indicators

Once some users have associated with an AP (e.g. when a train arrives near a platform or passengers arrive to board a train), each such user may start one (or more) app(s) in few tens of seconds (or may be already running in some cases). As this number of users changes at different locations in that cell, we use our ML models to predict change in aggregate throughput and RULI that this particular AP is expected to observe in near future. We now define, $w_i^{predict}(t)$, for use in Equation 2, using certain predicted parameters for client station i belonging to QoS class c at time t as follows:

$$w_i^{predict}(t) = \begin{cases} ww_i^{predict}(t), \text{when this factor is active} \\ 1, \text{otherwise} \end{cases}$$

Here,

$$ww_i^{predict}(t) = AggrT\_n_i^{predict}(t, c(i)) * RULI\_n^{predict}(t, c(i)) \quad (3)$$

$$AggrT\_n_i^{predict}(t,c(i)) = \max\{(A\_\max thresh(c(i)),$$
$$\max\{\left(M\ *(AggrT^{predict}(t+\delta) - AggrT(t))\right)^{\lambda(c(i))}\}$$

$$RULI\_n^{predict}(t,c(i)) = \max\{R\_\max thresh(c(i)),$$
$$(L*(RULI^{predict}(t+\delta) - RULI(t)))^{\beta(c(i))}\}$$

Here, $AggrT^{predict}(t+\delta)$, is the predicted value of aggregate throughput and $RULI^{predict}(t,t+\delta)$ is the predicted value of RULI at time (t+δ) for δ > 0 for that AP. A_maxthreshold(c(i)) and R_maxthresh(c(i)) are pre-defined thresholds for class c(i) to limit the impact of respective factors on scheduling weight and c(i) is the QoS class of most demanding (e.g. in terms of latency / jitter constraints) application of client i. We use L and M as normalizing constants (or they could be dynamically decided). In addition, we use parameters, λ(c(i)) and β(c(i)), to control impact of predicted factors on overall weight factor that is used by the PF metric. For a client station i supporting delay sensitive applications, we use λ(c(i)) ≥ 0 and β(c(i)) ≥ 0. For a station j supporting only best effort or background applications, we use λ(c(i)) = 0 = β(c(i)). For two stations, j1 and j2, with each supporting delay sensitive applications though j1 supporting an application with more stringent delay constraints than station j2, we choose λ(c(j1)) ≥ λ(c(j2)) and β(c(j1)) ≥ β(c(j2)). If we have access to predicted change in aggregated throughput but do not have access to predicted value of RU Load Indicator, we set β = 0 (and vice versa). Thus, we can implement various resource allocation policies by choosing suitable values of these parameters (such as α and β) and deciding when to activate or de-activate this weight factor.

Note that we presented above for PF scheduler but above predicted parameters and weights can also be used by other QoS scheduler.

III. Prediction of Traffic Load and RU Load Indicator

A. A Railway Station Scenario (Adapted from a railway station timetable given on web)

We consider a railway Station scenario to explain our method here. In a railway station, there can be APs located at platforms, waiting rooms and other locations. We consider an AP that is located at a specific platform (say x1). There is another platform x2 that is parallel to this and there is a railway track between x1 and x2. Some passengers from platform x1 and x2 end up associating with this AP. Let's say that m trains arrive on platform x1 in a day. We consider a (crowding) factor, denoted as TF(t) at time t, and say that number of passengers near this AP are in proportion to this factor TF at that time. Let's say that a train arrives at time, t_arr, and departs that platform at time, t_dept. We choose this factor TF to be between 0.9 and 1.1 during t_arr and t_dept. In general, this factor, TF, may be very low when no train is there on the platform. We assume that passengers start coming to board the train 40 min before the train arrives and we keep increasing TF randomly during this 40 min period until the train arrives. Once the train has departed (and no new train arriving in next 20 min), we decrease this TF over a period of 20 min until TF reaches zero. Assumption is that the passengers for whom this is the final destination, have left the platform in 20 min. Note that TF equal to zero means low number of passengers and it need not be zero.

During t_arr and t_dept, some passengers may get down from the train (and use WLAN network), some may board the train and some may use WLAN while sitting in that train as this is just an intermediate stop for them. Some such passengers may download a movie during that time (say via Netflix) or some may watch a short video clip via YouTube or use some other internet services. If two trains are present near platform x1, one on a track on left and other on a track on right, this TF factor can become greater than 1.

As only a fraction of train passengers may actually end up using WLAN network, we use a random factor to find the number of users who actually associate with this AP. We distribute users in three areas of this cell – edge, middle and center randomly. At periodic intervals before train arrival, we pick up two (of these three location) zones randomly and allocate users to each these two zones. We do this allocation randomly but keep it in the range of 10 – 40 % of total users at that time. Remaining users are assigned to the third zone. When TF is zero (i.e. no train at the platform), we choose random number of users (in the range 40- 45%) to be at center, 30 – 35% in the middle zone and remaining in the edge zone. We also assume that a fraction of these users use delay sensitive apps (such as video streaming) and we choose this fraction also randomly. We use this generated data for y months, with interval between two observations to be configurable. For example, this interval could be 1 min or 30 sec or 10 sec (or could be dynamically decided based on some events). We generated this data for 129161 time instants for a period of 3 months.

We divide each day into three periods, P-I (12 am to 6 am), P-II (6 am to 9 pm) and P-III (9 pm to 11:59 pm). We assume that there may be lower aggregate traffic load via an AP during the period P-I when compared to period P-II or P-III. We start with certain level of traffic load and allow changing that during any given period. At the beginning of period P-I, we assume that each user close to AP is using (or getting) throughput in the range of 20 – 50 Mbps, 10 – 20 Mbps in the middle zone and 1 – 9 Mbps in the edge zone. For period P-III, we assume similar throughput per-user as in period P-I. We also increase this by a randomly chosen

factor in the range of 4% to 7% at certain time instants during this period (for users that stay associated with that AP). For period P-II, we assume that some users may be accessing higher throughput apps and we randomly choose values in the range of 100-150 Mbps for each close-by user, 20 – 80 Mbps for each central zone user and 1-9 for each edge user. We also allow to randomly change this for each user using a factor (randomly chosen between 0.9 and 1.2) at certain time instants during this period when they stay associated with that AP. We show sample data that we generate in Table 1. For parameter z, Δz shows change in parameter z per-time unit in this table.

We use two datasets here. Dataset I is generated using the method described above. Dataset II is also generated using similar method but it is assumed that we do not have location information of users available at the time of association and that location information is not used for generating data. Scatter plots for some of the input variables are shown from Figure 2 to Figure 5 for dataset I and II.

Table 1: Sample Data

Time stamp (in time-unit)	N, Nds, AggrT	ΔN per time-unit, ΔNds per time-unit, ΔAggrT Mbps	(Ne, Nm, Nc) , (ΔNe(t), ΔNm(t), ΔNc(t))	HSLI, RULI
x=0 (start)	0,0,0	0,0,0	(0,0,0), (0,0,0)	--
x+1	15, 8, 300	15, 8, 300, 1	(2, 6, 7), (2, 6, 7)	L, L
x+2	19, 11, 350	+4, +3, + 50	(3, 8, 8), (1, 2, 1)	M, M
x+3	34, 21, 600	+15, +10, +250	(8, 14, 12), (5, 6, 4)	M, H
..	..	..	..	..

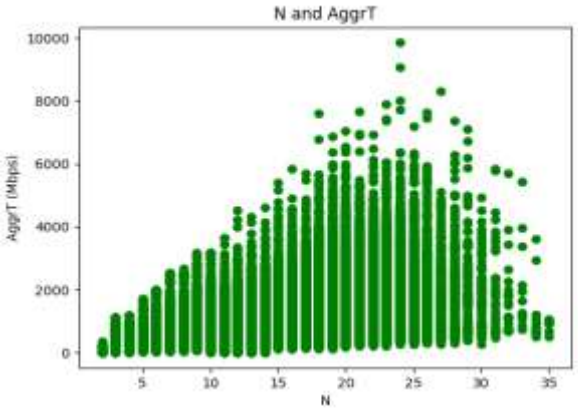

Figure 2: N and Aggregate Throughput for Dataset I

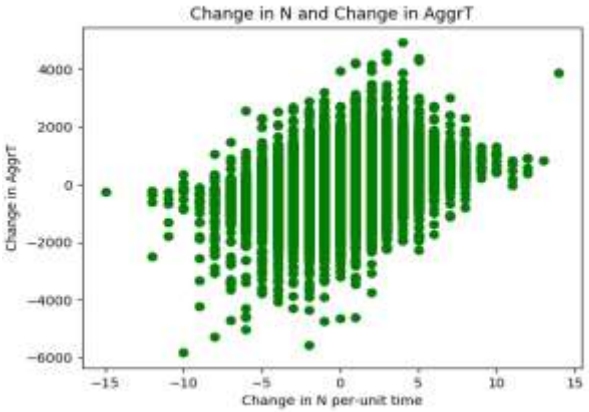

Figure 3: ΔN and $\Delta AggrT(t)$ for Dataset I

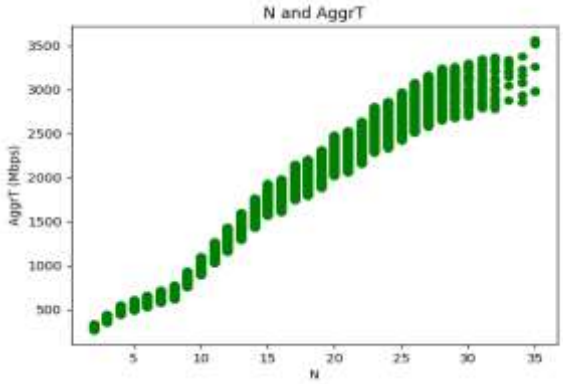

Figure 4: N and Aggregate Throughput for Dataset II

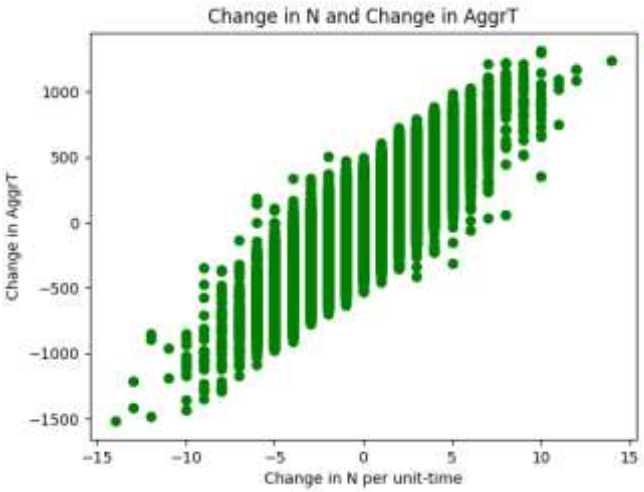

Figure 5: ΔN(t) and ΔAggrT(t) for Dataset II

B. ML Methods to predict traffic load and RULI:

We use ML methods such as SVM / SVR, KNN or Linear Regression to predict traffic load and RULI.

We allow use of following combinations of features for predicting aggregate throughout (or change in that). A suitable one can be selected depending on parameters that have been captured in a deployed scenario. If location data of a user has been captured, we recommend use of #8 below.

(Features) : (Target Variable)

1. (Number of users, N) : (Aggregate Throughput, i.e. AggrT at AP)

2. (Number of users at different location, i.e. Ne, Nm, Nc) : (AggrT)

3. (Net rate of change in N, i.e. deltaN) : (AggrT)

4. (deltaN, AggrT) : (Change in aggregate throughput, deltaAggrT)

5. (deltaN, N, AggrT) : (deltaAggrT)

6. (Net rate of change in users at different locations, i.e. deltaNe, deltaNm, deltaNc) : (deltaAggrT)

7. (deltaNe, deltaNm, deltaNc, AggrT) : (deltaAggrT)

8. (deltaNe, deltaNm, deltaNc, N, AggrT) : (deltaAggrT)

Similarly, RULI is predicted using above parameters. If we know type of apps (such as delay sensitive or non-delay sensitive), that information is also used to predict RULI.

C. Some examples using the railway station scenario given earlier

We now present ML methods for the problems described in the earlier section. We first apply linear regression method for different combinations of input variables to predict AggrT and ΔAggrT. Coefficient of determination, R-square, and Root Mean Square Error (RMSE) for different scenarios are given in Table 2. In L-2 in Table 2, we consider location of users and as expected, it helps to reduce RMSE compared to L-1 for dataset I. In L-4, we consider current number of associated users and throughput, in addition to change in throughput (as in L-3), and it slightly helps to reduce error (over L-3) for dataset I as well as II. Rate of addition of users (as in L-3) may be same at 7 am and 7 pm on a given day but number of users who watch a movie (via video streaming) at 7 pm may be different than that number at 7 am. Addition of input variables, N(t) and AggrT(t) as in L-5, helps to capture this impact if it exists. L-6 captures impact due to location of users and reduces RMSE for dataset I significantly. Adding N as a feature in L-8 doesn't help much over L-7 as L-7 already includes AggrT and other dominant features for this scenario. RMSE for ΔAggrT goes down from L-3 to L-8. Results for some of these scenarios with KNN (for K=6) are given in Table 3. Results for some scenarios for SVR with linear and RBF/Polynomial kernels are given in Table 4 and

Table 5 respectively. For SVR-polynomial kernel, we get lowest RMSE with K=1 and this resembles results that we get with SVR (Linear) as in SL-3. RMSE for some scenarios is compared in Figure 6. Note that we use 10-fold cross validation for the ML methods here and we used Python / SciKit for our ML work here. Even though RMSE of approximate 100 - 200 Mbps may look somewhat on higher side if one considers it in absolute sense, it still works well for our framework as we control impact of each term in equation (3) while computing scheduling weights using parameters like λ and maximum threshold values.

Table 2: Linear Regression for Dataset (DS) I and II with 10-fold cross-validation

	(Features), Target variable	RMSE (Mbps)		R-Square	
		DS I	DSII	DS I	DSII
L-1	(N), AggrT	577	113	0.41	0.97
L-2	(Ne, Nm, Nc), AggrT	524	NA	0.51	NA
L-3	(ΔN), ΔAggrT	370	129	0.14	0.69
L-4	(ΔN, AggrT), ΔAggrT	359	128	0.19	0.70
L-5	(ΔN, N, AggrT), ΔAggrT	350	106	0.22	0.79
L-6	(ΔNe, ΔNm, ΔNc), ΔAggrT	217	NA	0.70	NA
L-7	(ΔNe, ΔNm, ΔNc, AggrT), ΔAggrT	214	NA	0.71	NA
L-8	(ΔNe,ΔNm,ΔNe, N, AggrT), ΔAggrT	212	NA	0.71	NA

Table 3: KNN for Dataset (DS) I and II (with 10-fold cross validation)

	(Features), Target variable	RMSE (Mbps)		R-Square	
		DS I	DS II	DS I	DS II
N-1	(N), AggrT	619	90	0.32	0.98
N-2	(Ne, Nm, Nc), AggrT	556	NA	0.45	NA
N-3	(ΔN, N, AggrT), ΔAggrT	367	102	0.15	0.81
N-4	(ΔNe,ΔNm,ΔNc) , ΔAggrT	232	NA	0.66	NA
N-5	(ΔNe,ΔNm,ΔNc, AggrT), ΔAggrT	222	NA	0.68	NA

Table 4: SVR Linear (SL) Kernel for Dataset (DS) I & II

	(Features), Target variable	RMSE (Mbps)		R-Square	
		DS I	DS II	DS I	DS II
SL-1	(N), AggrT	581	115	0.40	0.97
SL-2	(Ne, Nm, Nc), AggrT	525	NA	0.51	NA
SL-3	(ΔN), ΔAggrT	372	129	0.12	0.69
SL-4	(ΔNe,ΔNm,ΔNc), ΔAggrT	218	NA	0.69	NA

Table 5: SVR with Polynomial and RBF Kernels with Dataset I

SVR Kernel	(Features), Target variable	RMSE (Mbps)
RBF	(ΔNe,ΔNm,ΔNc), ΔAggrT	235
Polynomial	(ΔN), ΔAggrT	378 (for K=1)

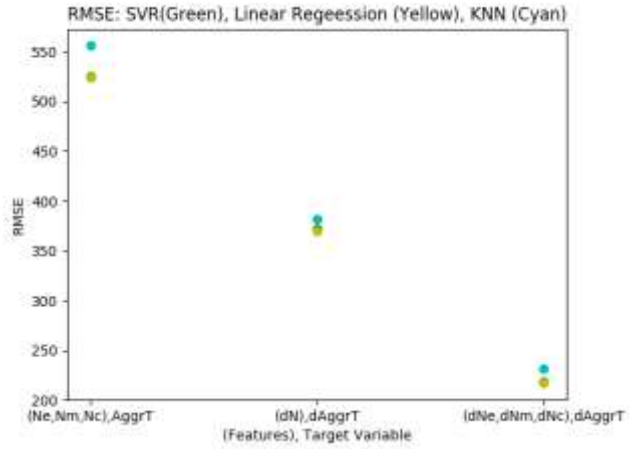

Figure 6: RMSE for dataset I (dz refers to Δz for z = Ne, Nm, Nc, N and AggrT in this plot)

We now present sample numerical results where we apply above techniques for OFDMA DL/UL QoS scheduling mechanisms. We use a simulator where we have simulated OFDMA MAC layer (and abstracted physical layer) for 802.11ax-type AP. We first consider a scenario where N(t)=20, channel bandwidth = 20 MHz and number of video streaming users = 10. All users are located in the range of 1 – 15 m from the AP. The AP monitors features as in L-6 of Table 2 or SL-4 of Table 4. It predicts change in throughput that it will need to support one time-unit from the time when some users join this AP. It starts adjusting its weights (as in equation 3) by $w_i^{predict}(t)$ in one time-unit. In Figure 7, we show impact on DL throughput, with and without such predicted

information, for video streaming users. We show impact of such predicted information on UL performance in Figure 8. We find that throughput improves for several existing (video) users in this scenario with ML approaches. It happens as we predict certain parameters as users join this AP and give improved service to some users during this transition period.

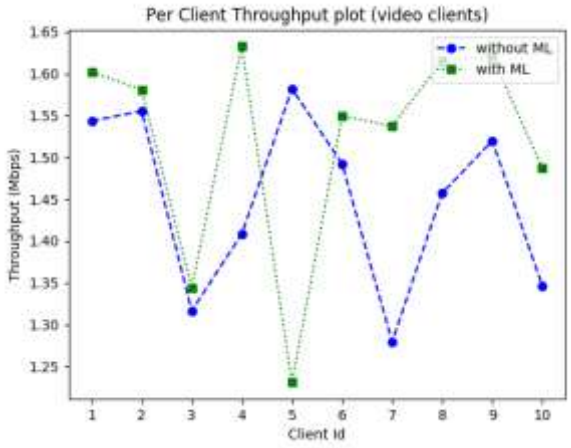

Figure 7: DL Throughput for video streaming users with enhanced proportional fair QoS scheduler

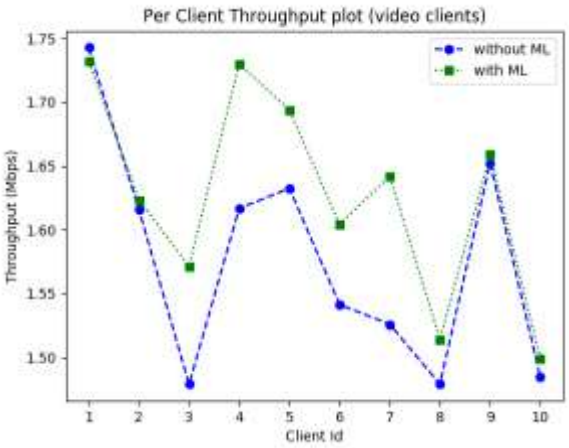

Figure 8: UL Throughput for Video users

Note that we presented above for PF scheduler but above predicted parameters and weights can also be used by other resource allocation methods for 802.11ax / OFDMA and other similar systems.